

Verbkonjugation im Lexikonformalismus *LEX*

Johannes Heinecke
Gunter Gebhardi

Humboldt Universität zu Berlin

Februar 1995

Johannes Heinecke
Gunter Gebhardi

Computerlinguistik
Institut für deutsche Sprache und Linguistik
Philosophische Fakultät II
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin

Tel.: (030) 20192 - 553

Fax: (030) 20192 - 560

e-mail: {heinecke|gebhardi}@compling.hu-berlin.de

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Forschung und Technologie (BMFT) unter dem Förderkennzeichen 01 IV 101 G gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei den Autoren.

Inhaltsverzeichnis

1	Überblick und Ausgangspunkt	2
2	Klassifizierung der Flexionsparadigmen	2
2.1	Regelmäßige Verben	3
2.2	Unregelmäßige Verben	4
3	Implementation	5
3.1	Der Eintrag im $\mathcal{L}\mathcal{A}\mathcal{V}$ -Basefile	5
3.2	Eintrags- <i>classes</i>	6
3.3	Flexions- <i>classes</i>	6
3.4	Die Treibermakros	7
3.5	Die Suffigierungsmakros	8
4	Ausblick	12

1 Überblick und Ausgangspunkt

Der folgende Report soll die Implementation der Verbkonjugation im VERBMOBIL-Lexikon verdeutlichen. Sie wurde im Lexikon-Formalismus $\mathcal{L}\mathcal{V}$ ¹ implementiert und soll hier vorgestellt werden.

Im VERBMOBIL-Projekt ist das Lexikon eine zentrale Wissensbasis für die syntaktische und semantische Analyse des gesprochenen Eingabesatzes. Hier wird die gesamte Information, die von dem Lexem (inkl. seiner flektierten Formen) alleine, d.h. ohne kontextabhängige Funktionen und Bedeutungen, kompakt kodiert, auf die alle linguistischen Analysekomponenten Zugriff haben. Dieses Lexikon ist mit Vollformen aufgebaut. Um zu gewährleisten, daß bei flektierbaren Lexemen (in der Regel, Nomina², Adjektive und Verben³) die Information, die nicht flexionsdependent ist, nicht von einer Flexionsform zur anderen variiert (z.B. durch inkonsistente Dateneingabe, bzw. um regelbasierte Informationen aus ökonomischen Gründen automatisch generieren zu können, damit letztendlich die Eingabedatenbank klein bleibt, ohne die Leistungsfähigkeit des Lexikons zu schmälern), wurden für flektierbare Lexeme Regeln zur maschinellen Erzeugung von Oberflächenformen implementiert.

Von der automatischen Konjugation vorerst ausgeschlossen bleiben vorerst alle Modal- und Controlverben, da sie sowohl syntaktisch als auch semantisch einer speziellen Beschreibung bedürfen. Die Konjugation (vom Auxiliar *sein* einmal abgesehen) gehört zwar in die unten erläuterten Paradigmen, die Klassenhierarchie von $\mathcal{L}\mathcal{V}$ setzt jedoch eigene Flexionsmechanismen voraus, auf die zu implementieren vorerst verzichtet wurde.

Aus gleichen Erwägungen werden zur Zeit auch keine Konjunktivformen und Imperative erzeugt, da deren Bedarf im VERBMOBIL-Projekt sich nur auf wenige Verben erstreckt. Eine Erweiterung ist jederzeit möglich.

2 Klassifizierung der Flexionsparadigmen

Die am folgenden beschriebenen Kriterien der Klassifizierung der Paradigmen genügen nicht unbedingt einer durch und durch linguistisch motivierten Einteil-

¹Lexikonformalismus, cf. [2]

²Cf. [3].

³Die Erzeugung von Ordinalzahlen (mit Flexionsformen) aus den Kardinalzahlformen muß ebenfalls hierzu gezählt werden.

lung. Vielmehr wurde ein pragmatischer Mittelweg zwischen linguistisch motivierten Klassen und Implementationsaufwand gegangen. So ist der Ablaut bei einigen unregelmäßigen Verben in der 2. und 3. Person Singular Präsens (*halten; du hältst, er hält* durchaus regelbasiert⁴, jedoch stellt sich bei der Frage, ob die Implementation eines „Ablaut-Generator“ der individuellen Angabe des umgelauteten Stamms im Basiseintrag vorzuziehen ist, die Frage, ob sich der Aufwand der Implementation für eine beschränkte Gruppe von (unregelmäßigen) Verben lohnt.

2.1 Regelmäßige Verben

Die regelmäßigen⁵ Verben zeichnen sich bekanntermaßen dadurch aus, daß sie einen unveränderlichen Stamm haben (kein Ablaut), jedoch in Abhängigkeit des Stammauslauts verschiedene Erweiterungen bzw. Kürzungen (-e-) annehmen.

In Abhängigkeit dieser verschiedenen Suffixe sind die regelmässigen Verben in (zur Zeit) sieben Konjugationsklassen eingeteilt, wie die folgende Übersicht verdeutlicht:⁶

	Beispiel	Präsensendungen					Präteritalendungen				Part. 2	Inf.
		1sg	2sg	3sg	1/3pl	2pl	1/3sg	2sg	1/3pl	2pl		
r1)	loten	-e	-est	-et	-en	-et	-ete	-etest	-eten	-etet	-et	-en
r2)	sage	-e	-st	-t	-en	-t	-te	-test	-ten	-tet	-t	-en
r2p)	trainieren, überlegen	-e	-st	-t	-en	-t	-te	-test	-ten	-tet	-t ^{a)}	-en
r3)	sammeln, ändern	-e	-st	-t	-n	-t	-te	-test	-ten	-tet	-t	-n
r4)	rasen, feixen, hetzen	-e	-t	-t	-en	-t	-te	-test	-ten	-tet	-t	-en
r5)	passen	-e	-t ^{b)}	-t ^{b)}	-en	-t ^{b)}	-te ^{b)}	-test ^{b)}	-ten ^{b)}	-tet ^{b)}	-t ^{b)}	-en
r6)	zeichnen	-e	-st	-t	-en	-t	-ete	-etest	-eten	-etet	-et	-en

Tabelle 1: Konjugationsklassen der regelmäßigen Verben in $\mathcal{L}\mathcal{V}$

^{a)} Das Partizip 2 wird ohne das Präfix **ge-** gebildet.

^{b)} Stammauslautendes **-ss** wird zu **-ß**.

⁴Cf. [1] S.125, der die reinen Konjugationssuffixe (ohne Berücksichtigung des Stammauslauts) tabellarisch auflistet, andere ([4] S. 20) listen mehr oder wenig komplette Paradigmen für alle unregelmäßigen Verben auf.

⁵[1] S. 115, fn. 1 spricht neuerdings nicht mehr von „starken“ und „schwachen“ Verben, sondern, da diese traditionellen Begriffe irreführend sind, von „unregelmäßigen“ und „regelmäßigen“ Verben

⁶Cf. [1] S.115ff.

Diese Konjugationsklassen wurden in Abhängigkeit der Verben, die in der Demonstratorwortliste vertreten sind gebildet. Zeigt sich bei der Vergrößerung des Lexikons die Notwendigkeit weiter Konjugationsmuster aufzunehmen, so kann dies mit geringstem Aufwand in $\mathcal{L}\mathcal{V}$ bewerkstelligt werden.

2.2 Unregelmäßige Verben

Um den Ablaut der unregelmäßigen Verben in den konjugierten Formen (2/3sg Präsens, Präteritum, Partizip 2) korrekt bilden zu können, werden in den Einträgen die abgelauteten Stämme mit angegeben. Da – wie oben schon angedeutet – der Implementationsaufwand für die Ablautregeln in keinem Verhältnis zu der Häufigkeit der Ablautreihen steht.

	Beispiel	Präsens					Präteritum				Part. 2
		1sg	2sg	3sg	1/3pl	2pl	1/3sg	2sg	1/3pl	2pl	
ir1)	gehen, singen, bleiben	-e	-st	-t	-en	-t	-	-st	-en	-t	-en
ir1p)	umerziehen, verschieben	-e	-st	-t	-en	-t	-	-st	-en	-t	-en ^{a)}
ir2)	denken, bringen	-e	-st	-t	-en	-t	-e	-est	-en	-et	-
ir3)	halten	-e	-st	-t	-en	-et	-	-est	-en	-et	-en
ir4)	finden, binden	-e	-est	-et	-en	-et	-	-est	-en	-et	-en
ir5)	lesen messen,	-e	-t	-t	-en	-t	-	-est	-en	-et	-en
ir6)	essen	-e	-t	-t	-en	- ^{b)}	-	-est	-en	-t	-en
ir6p)	vergessen	-e	-t	-t	-en	- ^{b)}	-	-est	-en	-t	-en ^{a)}
ir7)	reißen	-e	-t	-t	-en	-t	- ^{b)}	-est	-en	-et	-en
ir8)	wissen	-	-t	-t	-en	-t ^{b)}	-e	-est	-en	-et	-
ir9)	tun	-e	-st	-t	-n	-t	-	-st	-en	-et	-n

Tabelle 2: Konjugationsklassen der regelmäßigen Verben in $\mathcal{L}\mathcal{V}$

^{a)} Das Partizip 2 wird ohne das Präfix **ge-** gebildet.

^{b)} Stammauslautendes **-ss** wird zu **-ß**.

3 Implementation

Da die Konjugationsregeln der regelmäßigen Verben in $\mathcal{L}\mathcal{V}$ eine Teilmenge derjenigen Regeln für die unregelmäßigen Verben sind, stellen wir hier exemplarisch die Flexion der unregelmäßigen Verben dar. Der Unterschied zwischen den beiden Verbarten besteht in $\mathcal{L}\mathcal{V}$ darin, daß die unregelmäßigen Verben in ihrem Grundeintrag mehr Stammformen benötigen, da das Ablautphänomen aus oben bereits erwähnten Gründen nicht regelhaft erfaßt ist.

Verben mit unbetonten Präfixen erhalten kein *ge-* Präfix im Partizip 2. Aus diesem Grunde werden wir zwei Beispielverben vorstellen (*sprechen* und *verschieben*, um die Flexionsbildungsregeln zu erklären.

Der Aufbau der Flexionsgenerierung entspricht diesem Schema

- *base*-Eintrag (3.1)
- Eintrags-*classes* (3.2)
- Flexions-*classes*(3.3)
- Treibermakros (3.4)
- Suffigierungsmakros (3.5)

3.1 Der Eintrag im $\mathcal{L}\mathcal{V}$ -Basefile

Ausgegangen wird von dem Eintrag im Lexikon-Basefile⁷, in dem die wortspezifischen Informationen stehen, die nicht regelbasiert abgeleitet werden können:

```
base sprechen :<< intr_verben_c >>:      %% LeX4-class
    konj_klasse: ir1 &                    %% Konjugationsklasse
    praesstamm1: sprech &                 %% Abgelautete Staemme
    praesstamm2: sprich &
    praetstamm: sprach &
    partstamm: sproch .
```

```
base verschieben :<< tr_verben_c >>:
```

⁷Die Termini *base* und *class* werden im folgenden als *termini technici* des Lexikonformalismus $\mathcal{L}\mathcal{V}$ verwendet und sind wie in [2] dokumentiert zu verstehen.

```

konj_klasse: ir1p &
praesstamm1: verschieb &
praesstamm2: verschieb &
praetstamm: verschob &
partstamm: verschob &
syn_ibm: d_objekt: nonlex_case .

```

3.2 Eintrags-classes

Weiteren Information zieht $\mathcal{L}\mathcal{V}$ aus den Eintrags-classes, die die Merkmale bestimmen, die im *base*-Eintrag bereits gesetzt werden, und spezifische Werte des individuellen Verbs beinhalten:

```

class verb_staemme_c :< top >:
  subjekt: nonlex_case &
  konj_klasse: top &
  praesstamm1: top &
  praesstamm2: top &
  praetstamm: top &
  partstamm: top &
  restrict: restrict_v .

class intr_verben_c :< verb_staemme_c >: %% Intransitive Verben
  subjekt: nonlex_case .

class tr_verben_c :< verb_staemme_c >: %% Transitive Verben
  syn_ibm: d_objekt: (nonlex_case \ '^gen' \ dat \
                    pp: top \ cp: top) .

```

3.3 Flexions-classes

Die eigentlichen Flexionsregeln werden von den Flexions-classes aufgerufen, die ihrerseits alle Informationen der Eintrags-classes erben. Diese sind nach der Valenz der Verben und ob diese abtrennbare Präfixe besitzen unterteilt. Weiterhin sind die classes für die regelmäßigen Verben ein wenig simpler, da diese unter Angabe eines einzigen Stammes flektiert werden können. Hier sind nur die für die exemplarische Behandlung der Konjugation notwendigen classes wiedergegeben:

```

%% finite Verbformen von intransitiven unregelmässigen Verben
class intr_fin_v_syn_c :< intr_verben_c >:
    VERB &
    #ir_verb_flexion(VERB) &
    syn_ibm: ibm_macro: intrans_fin_syn .

%% infinite Verbformen von intransitiven unregelmässigen Verben
class intr_infin_v_syn_c :< intr_verben_c >:
    VERB &
    #ir_infinite_formen(VERB) &
    syn_ibm: ( ibm_macro: intrans_infin_syn &
               status: status_v ) .

...

%% finite Verbformen von transitiven unregelmässigen Verben
class tr_fin_v_syn_c :< tr_verben_c >:
    VERB &
    #ir_verb_flexion(VERB) &
    syn_ibm: ibm_macro: trans_fin_syn .

%% infinite Verbformen von transitiven unregelmässigen Verben
class tr_infin_v_syn_c :< tr_verben_c >:
    VERB &
    #ir_infinite_formen(VERB) &
    syn_ibm: (status: status_v &
               ibm_macro: trans_infin_syn ) .

```

Der gesamte *base*-Eintrag des jeweiligen Verbs wird an die Variable `VERB` gebunden, die dann den entsprechenden Flexionsmakros übergeben wird. Da die Flexion natürlich von der Valenz völlig unabhängig ist, werden die selben Makros verwendet.

3.4 Die Treibermakros

Zuerst wird das Treibermakro aufgerufen, die anhand der Konjugationsklasse (in `konj_klasse`: im *base*-Eintrag angegeben) die Suffigierungsmakros aktiviert:

Die Trennung der Treibermakros nach finiten und infiniten Formen hat seine Ursache in den Unterschiedlichen Erfordernissen dieser Formen sowohl in Syntax und Semantik, die im Ausgabe-Vollformenlexikon (STUF-III) in verschiedenen *templates* erscheinen. Da diese STUF-*template*-Namen bereits in den Flexions-*classes*

kodiert werden, kann durch verschiedene Treibermakros verhindert werden, daß finite Verbformen mit den *STUF-templates* für infinite Formen und umgekehrt generiert werden.

```
#ir_verb_flexion(Verb) :: %% Finite Formen
    Verb &
    konj_klasse: K &
    #verb_klasse(K, Verb) .

#ir_infinite_formen(Verb) :: %% Infinite Formen
    Verb &
    konj_klasse: K &
    #inf_verb_klasse(K, Verb) .
```

3.5 Die Suffigierungsmakros

Die obigen Beispieleinträge gehören den Konjugationsklassen *ir1* bzw. *ir1p* an⁸. Der folgende Ausschnitt zeigt die Implementation der Suffigierungsmakros dieser beiden Klassen:

```
% Klasse ir1) sprechen
#verb_klasse(ir1, Verb) :: #sg_1_pres(Verb, e) .
#verb_klasse(ir1, Verb) :: #sg_2_pres(Verb, st) .
#verb_klasse(ir1, Verb) :: #sg_3_pres(Verb, t) .
#verb_klasse(ir1, Verb) :: #pl_1_3_pres(Verb, en) .
#verb_klasse(ir1, Verb) :: #pl_2_pres(Verb, t) .

#verb_klasse(ir1, Verb) :: #sg_1_3_pret_ne(Verb) .
#verb_klasse(ir1, Verb) :: #sg_2_pret(Verb, st) .
#verb_klasse(ir1, Verb) :: #pl_1_3_pret(Verb, en) .
#verb_klasse(ir1, Verb) :: #pl_2_pret(Verb, t) .

#inf_verb_klasse(ir1, Verb) :: #part(Verb, en) .
#inf_verb_klasse(ir1, Verb) :: #inf(Verb, en) .
#inf_verb_klasse(ir1, Verb) :: #infzu(Verb, en) .
```

```
% Klasse ir1p) verschieben
```

⁸Cf. Tabelle 2 auf Seite 4

```

#verb_klasse(ir1p, Verb) :: #sg_1_pres(Verb, e) .
#verb_klasse(ir1p, Verb) :: #sg_2_pres(Verb, st) .
#verb_klasse(ir1p, Verb) :: #sg_3_pres(Verb, t) .
#verb_klasse(ir1p, Verb) :: #pl_1_3_pres(Verb, en) .
#verb_klasse(ir1p, Verb) :: #pl_2_pres(Verb, t) .

#verb_klasse(ir1p, Verb) :: #sg_1_3_pret_ne(Verb) .
#verb_klasse(ir1p, Verb) :: #sg_2_pret(Verb, st) .
#verb_klasse(ir1p, Verb) :: #pl_1_3_pret(Verb, en) .
#verb_klasse(ir1p, Verb) :: #pl_2_pret(Verb, t) .

#inf_verb_klasse(ir1p, Verb) :: #part_np(Verb, en) .  %% kein 'ge-'
#inf_verb_klasse(ir1p, Verb) :: #inf(Verb, en) .
#inf_verb_klasse(ir1p, Verb) :: #infzu(Verb, en) .

```

In einem letzten Schritt werden die Verbalendungen an die entsprechenden Stämme suffigiert, dabei werden etwaige Konsonantenveränderungen (z.B. *ss* > *ß*, wie in *essen*, *ihr eßt*) o.ä.) mitberücksichtigt. Die Variable *Verb* enthält immer noch den kompletten *base*-Eintrag, so daß die Suffigierungsmakros die jeweils benötigten Verbalstämme verwenden können. Die hier zum Teil aufgerufen Makros des Typs *#atommmm* sind *built-in*-Makros des Lexikonformalismus $\mathcal{L}\mathcal{V}\mathcal{4}$.

```

#sg_1_pres(Verb, Endung) ::
  Verb &
  restrict: sg1pres &
  praesstamm1: Stamm &
  syn_ibm: phon: #atomcat(Stamm, Endung) &
  person: 1 &
  tense: present_indicative &
  number: sg .

#sg_2_pres(Verb, Endung) ::
  Verb &
  restrict: sg2pres &
  praesstamm2: Stamm &
  syn_ibm: phon: #atomcat(Stamm, Endung) &
  person: 2 &
  tense: present_indicative &
  number: sg .

#sg_3_pres(Verb, Endung) ::
  Verb &

```

```
restrict: sg3pres &
praesstamm2: Stamm &
syn_ibm: phon: #atomcat(Stamm, Endung) &
person: 3 &
tense: present_indicative &
number: sg .

#pl_1_3_pres(Verb, Endung) :: %% Disjunktion, da formgleich
Verb &
restrict: pl1_3pres &
praesstamm1: Stamm &
syn_ibm: phon: #atomcat(Stamm, Endung) &
person: '^2' &
tense: present_indicative &
number: pl .

#pl_2_pres(Verb, Endung) ::
Verb &
restrict: pl2pres &
praesstamm1: Stamm &
syn_ibm: phon: #atomcat(Stamm, Endung) &
person: 2 &
tense: present_indicative &
number: pl .

#pl_2_pres_ss2sz(Verb, Endung) :: %% 'ss' am Stammende
Verb & %% wird abgeschnitten
restrict: pl2pres &
praesstamm1: Stamm &
tmp: (#atomsplz(Stamm, ss) & Temp) &
syn_ibm: phon: #atomcat(Temp, Endung) - tmp &
person: 2 &
tense: present_indicative &
number: pl .
```

Die Suffigierungsmakros für die Präteritalformen sind analog zu den Präsensformen zu verstehen. Etwas anders sehen die Suffigierungsmakros der infiniten Formen aus:

```
%% Partizip 2
#part(Verb, Endung) ::
Verb &
restrict: part2 &
```

```

partstamm: Stamm &
tmp: (#atomcat(ge, Stamm) & Temp) &
syn_ibm: ( phon: #atomcat(Temp, Endung) &
           status: part2 ) - tmp &
tense: nonfinite .

#part_np(Verb, Endung) ::                               %% Kein Pr''afix 'ge'
  Verb &
  restrict: part2 &
  partstamm: Stamm &
  syn_ibm: ( phon: #atomcat(Stamm, Endung) &
            status: part2 ) &
  tense: nonfinite .

%% Infinitiv
#inf(Verb, Endung) ::
  Verb &
  restrict: inf &
  praesstamm1: Stamm &
  syn_ibm: ( phon: #atomcat(Stamm, Endung) &
            status: infin ) &
  tense: nonfinite .

%% 'zu'-Infinitiv
#infzu(Verb, Endung) ::
  Verb &
  restrict: zu_inf &
  praesstamm1: Stamm &
  syn_ibm: ( phon: #atomcat(Stamm, Endung) &
            status: zu_inf ) &
  tense: nonfinite .

```

Die nun erzeugte Vollform wird in den Ausgabebereich geschrieben, und kann mit den semantischen Informationen verknüpft werden.

4 Ausblick

Diese Implementation ist in einer etwas erweiterten Form Teil des VERBMOBIL-Lexikons. Die modular gehaltene Implementation kann problemlos erweitert werden, wenn beispielsweise ein Verb spezielle, abweichende, Formen besitzt, die im momentan existenten VERBMOBIL-Lexikon noch nicht integriert worden sind. Neue Konjugationsklassen, bzw. eine Subdifferenzierung der bestehenden Konjugationsklassen, bedingt durch die Berücksichtigung der z.Zt. nicht generierten Konjunktivformen, kann ohne daß dazu eine aufwendig Neuimplementation bestehenden Codes notwendig wird, definiert werden.

Literatur

- [1] Der Duden in 12 Bänden. Bd. 4, Grammatik. Dudenverlag: Mannheim ⁴1984.
- [2] Gunter Gebhardi, Johannes Heinecke: Lexikonformalismus $\mathcal{L}\mathcal{V}$ — Sprachbeschreibung. Verbmobil Technisches Dokument 19. Humboldt-Universität zu Berlin 1995.
- [3] Gunter Gebhardi, Johannes Heinecke: Substantivflexion in $\mathcal{L}\mathcal{V}$ — Ein Applikationsbericht. Verbmobil Memo 62. Humboldt-Universität zu Berlin 1995.
- [4] Wahrig-Burfeind, R. (Hrsg.): Deutsches Wörterbuch. Bertelsmann Lexikon-Verlag: Gütersloh. 1994.